



Deconstructing stem cell tumorigenicity: a roadmap to safe regenerative medicine.

Journal: Stem Cells

Publication Year: 2009

Authors: Paul S Knoepfler

PubMed link: 19415771

Funding Grants: Molecular mechanisms governing hESC and iPS cell self-renewal and pluripotency

**Public Summary:** 

## Scientific Abstract:

Many of the earliest stem cell studies were conducted on cells isolated from tumors rather than from embryos. Of particular interest was research on embryonic carcinoma cells (EC), a type of stem cell derived from teratocarcinoma. The EC research laid the foundation for the later discovery of and subsequent work on embryonic stem cells (ESC). Both ESC isolated from the mouse (mESC) and then later from humans (hESC) shared not only pluripotency with their EC cousins, but also robust tumorigenicity as each readily form teratoma. Surprisingly, decades after the discovery of mESC, the question of what drives ESC to form tumors remains largely an open one. This gap in the field is particularly serious as stem cell tumorigenicity represents the key obstacle to the safe use of stem cell-based regenerative medicine therapies. Although some adult stem cell therapies appear to be safe, they have only a very narrow range of uses in human disease. Our understanding of the tumorigenicity of human induced pluripotent stem cells (IPSC), perhaps the most promising modality for future patient-specific regenerative medicine therapies, is rudimentary. However, IPSC are predicted to possess tumorigenic potential equal to or greater than that of ESC. Here, the links between pluripotency and tumorigenicity are explored. New methods for more accurately testing the tumorigenic potential of IPSC and of other stem cells applicable to regenerative medicine are proposed. Finally, the most promising emerging approaches for overcoming the challenges of stem cell tumorigenicity are highlighted.

Source URL: https://www.cirm.ca.gov/about-cirm/publications/deconstructing-stem-cell-tumorigenicity-roadmap-safe-regenerative-medicine